REMARKS

By this amendment, the specification has been editorially amended and claims 5, 7, 9 and 11 have been amended. Currently, claims 1-11 are pending in the application.

The indication that claims 1-4, 6, 8 and 10 are allowed is noted with appreciation.

The Examiner stated that the title of the invention was not descriptive. By this amendment, the title "SIGNAL PROCESSING UNIT AND SIGNAL PROCESSING METHOD" has been amended to "SIGNAL PROCESSING UNIT AND SIGNAL PROCESSING METHOD INCLUDING USING AN EXPONENT PART AND A MANTISSA PART FOR POWER GENERATION". It is respectfully submitted that the title is now descriptive of the invention. If the Examiner believes that a different title is more appropriate, the Examiner is respectfully requested to suggest this different title.

Claims 5, 7, 9 and 11 were rejected under 35 USC 102(b) as being anticipated by Lucas. The Examiner believed that Fig. 3 in Lucas and the abstract disclosed an apparatus and a method for converting floating point data to an integer by extracting bits of the exponent and the mantissa which are then used to address a

lookup table which in turn outputs the integer.

This rejection is respectfully traversed in view of the amendments to the claims and the following remarks.

The present invention relates to a signal processing unit and signal processing method for data in floating point format which calculate the p-th power of v (v^p), and to a program product for implementing the method.

The second embodiment of the present invention specifically discloses in Fig. 10, a block diagram showing the structure of an exponential calculation device. This signal processing unit comprises an exponent part and mantissa part extraction section 30 and a third conversion section 31. If the number of bits in which (N-2) is expressed in binary notation is M, the exponent part and mantissa part extraction section 31 is structured so as to extract a bit field consisting of at least the lowermost M bits of the exponent part and at least the uppermost (N-1) bits of the mantissa part. If the value given by the bit field which has been extracted from the exponent part and mantissa part extraction section 30 is termed w, the third conversion section 31 is structured so as to store in advance as a table the values of v converted to integers for all the addresses w which are

obtained for this bit field. The third conversion section uses the value w given by the bit field as an input and reads out the value from the table.

In the following, in order to simplify the explanation, the structure and operation of the signal processing unit which performs type conversion of a floating point value v to an integer value will be explained for the case when v is greater than or equal to 2.0 and less than 2^N, i.e. 32.0. In the exponent and mantissa part extraction section 30, a bit pattern which includes the uppermost N-1 bits of the mantissa part is extracted from the bit field which straddles the exponent part and the mantissa part of the inputted item of floating point data. In this second preferred embodiment the floating point format is supposed to be the one shown in Fig. 3, and the lowermost 2 bits of its exponent part and the uppermost 4 bits of its mantissa part are extracted. In other words, a continuous field from the 20th bit through the 25th bit of the floating point format is extracted, and this is termed w.

The intention of extracting this type of bit pattern is as follows.

At this time, the region for the floating point value v

which is inputted is greater than or equal to 2.0 and less than 32.0, and what is required is (int)(v^1). Accordingly, the output value is an integer greater than or equal to 2 and less than or equal to 31. Now here, for example, since the bit pattern for the floating point value 2.0 is:

0 10000000 00000000000000000000000

and the first pattern of the floating point value for 31.99999999999999 is:

0 10000011 11111111111111111111111.

Therefore in this region the bits that vary in the exponent part are only the lowermost two bits of said exponent part. To put it another way, it will be understood that, if these lowermost two bits are known, the process of adding 128 to their value can introduce the value of the original exponent part.

On the other hand, for the mantissa part, although all the 23 bits vary, since the effective number of digits in the final output value is 5 bits, the necessary bits for the mantissa part are only the uppermost 4 bits in the mantissa part. In other words, the bits of the mantissa part other than its uppermost 4 bits are bits which can be discarded from the point of view of the final result. Due to this, in the exponent and mantissa part

extraction section 30, only the minimum amount required is extracted, which consists of the lowermost 2 bits in the exponent part and the uppermost 4 bits in the mantissa part.

Independent claim 5 has been amended to recite "an exponent and mantissa part extraction section which, when the number of bits in which (N-2) is expressed in binary notation is M, extracts a bit field consisting of a predetermined number of lowermost M bits of said exponent part and a predetermined number of uppermost (N-1) bits of said mantissa part".

Similarly, independent claims 7, 9 and 11 have been amended to recite "extracting a bit field including a predetermined number of lowermost M bits of the exponent part and a predetermined number of uppermost (N-1) bits of the mantissa part when the number of bits in which (N-2) expressed in binary notation is M; and storing in a table in advance the values of v converted into integer values in all addresses where w is extracted by the bit field, and inputting the value w given by the bit field and reading out the corresponding value from the table when the value expressed by the bit field is w".

These features are not shown or suggested by the prior art of record.

Lucas relates to data processing systems having display apparatus and, in particular, to methods and apparatus for displaying graphical and other images.

Lucas discloses that in Fig. 2, the bits may be grouped together into two 16-bit "halfwords" and demonstrate that the high-order 16 bits of a single-precision floating-point number contain eight bits of precision, including the hidden high-order mantissa bit. It can therefore be realized that employing only the high-order halfword of an IEEE single-precision floating-point number introduces a representation error on the order of only 2^{-8}. For a typical pixel value conversion operation only eight bits of precision are available in the result (the one-byte pixel value) and thus the high-order 16 bits of the single-precision floating point number provide sufficient precision to represent all 256 of the possible eight bit pixel values.

Lucas also illustrates in Fig. 3 a discrete memory device implementation of the LUT (Lookup table) 24 of Fig. 1 and shows the coupling of the upper halfword of the single-precision floating-point number to the LUT 24 address (A) inputs.

Lucas also discloses that although floating-point values are generally 32-bits or more, there are a number of significant

applications where only a sub-set of the entire floating-point representation need to be employed as a table index value. In the presently preferred embodiment of Lucas, only 16-bits of the 32-bit single-precision floating-point value are relevant to the result; specifically the sign bit, the exponent, and a portion of the mantissa.

As described above, Lucas always requires the entire exponent part and lowermost portion of the mantissa part as a table index value. The present invention does not requires the above features of Lucas and the present invention only needs lowermost M bits of the exponent part and uppermost (N-1) bits of the mantissa part as a table index value.

Lucas does not disclose that an exponent and mantissa part extraction section which, when the number of bits in which (N-2) is expressed in binary notation is M, extracts a bit field consisting of a predetermined number of lowermost M bits of said exponent part and a predetermined number of uppermost (N-1) bits of said mantissa part as claimed in claim 5.

Lucas also does not disclose that extracting a bit field including a predetermined number of lowermost M bits of the exponent part and a predetermined number of uppermost (N-1) bits

of the mantissa part when the number of bits in which (N-2) expressed in binary notation is M; and storing in a table in advance the values of v converted into integer values in all addresses where w is extracted by the bit field, and inputting the value w given by the bit field and reading out the corresponding value from the table when the value expressed by the bit field is w as claimed in claims 7, 9 and 11.

Applicants respectfully submit Lucas does not suggest the claimed features as described above. Specifically, in the present invention, only a limited part of the inputted value is used for an address input to the third conversion section, and thereby the table memory can be minimized.

For these reasons, it is believed that Lucas does not show or suggest the present claimed features of the present invention. It is therefore submitted that claims 5, 7, 9 and 11 are allowable over Lucas.

In view of foregoing claim amendments and remarks, it is respectfully submitted that the application is now in condition for allowance and an action to this effect is respectfully requested.

Applicants also respectfully submit that the amendments to claims 5, 7, 9 and 11 were to clarify the claim language and were not done for reasons related to patentability.

If there are any questions or concerns regarding the amendments or these remarks, the Examiner is requested to telephone the undersigned at the telephone number listed below.

Respectfully submitted,

Date: October 12, 2004

Randolph A. Smith
Reg. No. 32,548

SMITH PATENT OFFICE
1901 Pennsylvania Ave., N.W.
Suite 200
Washington, DC 20006-3433
Telephone: 202/530-5900
Facsimile: 202/530-5902
Miyasaka101204